

**ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСІЛКИ
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»**

**ІНСТИТУТ ЕКОНОМІКИ, УПРАВЛІННЯ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
ФАКУЛЬТЕТ ЕКОНОМІКИ І МЕНЕДЖМЕНТУ
ФОРМА НАВЧАННЯ ДЕННА
КАФЕДРА МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ТА СОЦІАЛЬНОЇ
ІНФОРМАТИКИ**

Допускається до захисту

Завідувач кафедри _____ О.О. Ємець
(підпис)

«_____» _____ 2020 р.

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО ДИПЛОМНОЇ РОБОТИ**

на тему

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ТРЕНАЖЕРА З ТЕМИ «СОРТУВАННЯ
ХООРА» ДИСТАНЦІЙНОГО НАВЧАЛЬНОГО КУРСУ «АЛГОРИТМИ ТА
СТРУКТУРИ ДАНИХ»**

зі спеціальності 122 «Комп'ютерні науки»

Виконавець роботи Чипенко Богдан Юрійович
_____ «___» _____ 2020 р.
(підпис)

Науковий керівник канд. фіз.-мат. наук, доц. Ємець Олександра Олегівна
_____ «___» _____ 2020 р.
(підпис)

ПОЛТАВА 2020 р.

ЗМІСТ

| | |
|--|----|
| ВСТУП..... | 3 |
| 1 ПОСТАНОВКА ЗАДАЧІ..... | 5 |
| 1.1 Постановка задачі розробки навчального тренажера..... | 5 |
| 2 ІНФОРМАЦІЙНИЙ ОГЛЯД..... | 6 |
| 2.1 Актуальність дистанційного навчання в сучасності..... | 6 |
| 2.2 Метод Хоора «Швидке сортування»..... | 10 |
| 2.3 Огляд робіт, де розглянуто аналогічне до теми роботи завдання..... | 17 |
| 2.4 Позитивні аспекти оглянутих робіт..... | 18 |
| 2.5 Вади розробок з оглянутих робіт | 19 |
| 2.6 Необхідність та актуальність теми..... | 21 |
| 3 ТЕОРЕТИЧНА ЧАСТИНА..... | 22 |
| 3.1 Алгоритмізація задачі за темою..... | 22 |
| 3.2 Розробка блок-схеми, яка підлягає програмуванню..... | 26 |
| 3.3 Обґрунтування вибору програмних засобів для реалізації завдання роботи..... | 31 |
| 4 ПРАКТИЧНА ЧАСТИНА..... | 32 |
| 4.1 Опис процесу програмної реалізації..... | 32 |
| 4.2 Опис програми..... | 35 |
| 4.3 Перевірка валідності..... | 37 |
| ВИСНОВКИ..... | 51 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ..... | 53 |
| ДОДАТОК А. КОД ПРОГРАМА..... | 55 |

ВСТУП

Актуальність проведеної роботи заключається в реалізації тренажеру для підготовки студентів за програмою «Комп'ютерні науки». Студенти проходять підготовку за дисципліною «Алгоритми та структури даних», що містить тему «Сортування Хоора».

Нинішня ситуація в світі з карантинном та самоізоляцією показник - того, що дистанційне навчання повинно розвиватися та удосконалюватися. Завдяки дистанційному доступу до курсів та програм навчання в школах та вищих навчальних закладах нам вдалося продовжити та благополучно закінчити навчання 2020 року. На мою думку майбутнє за дистанційним навчанням, де фізичний контакт може бути зведений до мінімуму, бо бувають різні обставини та фізичні можливості людей. Інформаційні технології дають шанс майже всім займатися роботою або навчанням з будь-якого місця Земного шару, для цього потрібен тільки девайс для роботи. Тому розвиток сучасного світу шляхом інформатизації - це правильна ідея і вибір, бо і освіта зможе не стояти на місці, а рухатися з наукою пліч о пліч. Також дистанційне навчання може дозволити нам запобігти в майбутньому такому поширенню інших захворювань серед дітей та підлітків, та в майбутньому більш безболісно пройти період можливих пандемій.

Комп'ютеризація - є найперспективнішим напрямком інформатизації навчання вищих навчальних закладах. Інтенсивний розвиток обчислювальної техніки та методів програмування зробили великий крок за останні десять років у підготовці спеціалістів, у тому числі за допомогою комп'ютерних тренажерів.

Мета виконаної роботи полягає в алгоритмізації та створенні програми, за допомогою якої студенти будуть проходити підготовку за теми «Сортування Хоора» для дистанційного курсу «Алгоритми та структури даних».

Були поставленні наступні задачі:

1. Перегляд методичних рекомендацій та стандартів, для оформлення курсової роботи.
2. Формулювання постановки задачі роботи.
3. Пошук та вивчення інформації, що відноситься до «Сортування Хоора».

4. Інформаційний огляд досвіду з даної теми.
5. Розробка алгоритму.
6. Складання блок-схеми тренажеру.
7. Вибір мови програмування.
8. Створення програми для навчального тренажера.
9. Опис пояснювальної записки бакалаврської роботи.

Предметом розробки є навчальний тренажер для дисципліни «Алгоритми та структури даних» з теми «Сортування Хоора».

Навчальний тренажер було реалізовано в середовищі візуальної розробки програм Microsoft Visual Studio та об'єктно-орієнтована мова програмування C#.

Практична цінність цієї роботи полягає в підготовці студентів за темою «Сортування Хоора». Унікальність магістерської роботи полягає в створеному навчальному тренажері з теми «Сортування Хоора». Тренажеру з цієї теми не існує в мережі Internet.

Об'єктом магістерської роботи був навчальний тренажер – навчально-тренувальний засіб для відпрацювання навичок і вдосконалення знань. Технічний засіб професійної підготовки учня або студента, призначений для формування і вдосконалення професійних навичок і вмінь, необхідних для подальшого навчання або праці.

Ступінь готовності програми на тему «Сортування Хоора» дистанційного навчального курсу «Алгоритми та структури даних», на мою думку готова відсотків на 70, бо програма не ідеальна і покриті не всі навчальні нюанси цієї теми. До впровадження в навчальний курс не готова.

1 ПОСТАНОВКА ЗАДАЧІ

1.1 Постановка задачі розробки тренажера

Задачею магістерської роботи полягає в реалізації тренажеру для підготовки студентів за темою «Сортування Хоора» дистанційного навчального курсу «Алгоритми та структури даних». Необхідно слідувати вимогам, які зазначені для тренажеру.

Вимогами для розробки тренажера з теми «Сортування Хоора» для дистанційного курсу «Алгоритми та структури даних»:

1. Обрати мову програмування для створення тренажеру. Мова повинна бути сумісна з платформою Moodle.
2. Створити алгоритм тренажеру з теми «Сортування Хоора».
3. Реалізувати тренажер за допомогою обраної мови програмування .
4. Перевірити програму на наявність помилок.

Основні вимоги до навчального тренажеру:

1. Умова задачі повинна бути доступною для студента на кожному кроці.
2. Після вибору відповіді потрібно реалізувати перевірку даних. Якщо було допущено помилку, то треба повідомити користувача про неї.

2 ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1 Актуальність дистанційного навчання в сучасності

Останні кілька десятиріч років, навчання на базі комп'ютерних технологій витісняє свого попередника. Здебільшого це стало можливим саме через розвиток мережі Інтернет, що розв'язало руки у питанні зв'язку одного кінця світу з іншим, вільного розміщування інформації на Інтернет-сайтах, що дозволяло скористатися нею всім, кому вона потрібна.

Освітній процес вдосконалює свою ефективність за допомогою інформаційних технологій, які зараз активно розвиваються та проникають майже в кожну сферу сучасного життя.

Концепція дистанційної освіти в Україні — це форма навчання, що йде пліч о пліч з іншими формами навчання, що реалізується за технологіями дистанційного навчання.

Останні роки науково-методичні основи дистанційного навчання активно розвиваються, та реалізують в багатьох вищих навчальних закладах України. Не беручи до уваги великий обсяг наукових досліджень, сучасна дистанційна освіта в Україні більш нагадує форму заочного навчання, в якій не використовують всі можливості і методи.

На сьогоднішній день немає одної точки зору серед науковців, відносно «дистанційного навчання».

Дистанційне навчання — це технологія, що базується на принципах відкритого навчання, широко використовує комп'ютерні навчальні програми за допомогою сучасних телекомунікацій.

Дистанційне навчання має переваги перед більш традиційними формами навчання. За допомогою нього можна підтримувати регулярний контакт з викладачем та одержувати структурований навчальний матеріал, наданий в електронному вигляді. Але все одно незначна частина навчального процесу дистанційної освіти повинна здійснюватися за очною формою, а саме: складання іспитів, практичні і тому подібне.

Поєднання педагогічних та інформаційних технологій дають на виході технології дистанційного навчання. Характерні риси дистанційного навчання розглянуто в таблиці 2.1.

Дистанційне навчання надає доступ до нетрадиційних джерел інформації, підвищує продуктивність самостійної роботи, та дає абсолютно нові можливості для творчого самовираження. А в той час для викладачів відкривається новий шлях для реалізації абсолютно нових форм і методів навчання.

Дистанційне навчання розвиватиметься і вдосконалюватиметься із розвитком інтернет-технологій і вдосконаленням методів дистанційного навчання.

Таблиця 2.1 – Характерні риси дистанційного навчання

| Ознака | Суть |
|--------------------|---|
| Гнучкість | в основному ті хто користуються дистанційним навчанням не відвідують регулярних занять, а навчаються у зручний для себе час |
| Модульність | кожен курс моделює цілісне уявлення про окрему предметну область, що дозволяє з набору незалежних курсів-модулів сформувати навчальну програму |
| Паралельність | навчальний курс йде паралельно з професійною діяльністю або з навчанням в іншому місці |
| Велика аудиторія | одночасне звернення до багатьох джерел навчальної інформації великої кількості студентів |
| Економічність | ефективне використання навчального простору та технічних засобів, концентроване представлення інформації, знижують витрати на підготовку фахівців |
| Технологічність | використання в початковому процесі нових досягнень інформаційних технологій |
| Соціальна рівність | рівні можливості одержання освіти незалежно від місця проживання, стану здоров'я і соціального статусу |

| | |
|------------------------------|--|
| Інтернаціональність | можливість одержати освіту у навчальних закладах іноземних держав, не виїжджаючи зі своєї країни та надавати освітні послуги іноземним |
| Нова роль викладача | дистанційна освіта розширює і оновлює роль викладача, робить його наставником-консультантом, який повинен координувати пізнавальний процес |
| Позитивний вплив на студента | підвищення в людини потенціалу творчого та інтелектуального, що одержує дистанційну освіту, за |

| | |
|--------|--|
| | рахунок самоорганізації, використання сучасних інформаційних та телекомунікаційних технологій, вміння самостійно приймати відповідальні рішення |
| Якість | якість дистанційної освіти не поступається якості традиційній формі навчання, тому що для підготовки залучається найкращий професорсько-викладацький склад і використовується найсучасніші навчально-методичні матеріали |

2.2 Метод Хоора «Швидке сортування»

Метод Хоора «Швидке сортування», часто називають qsort (від імені стандартної бібліотеки мови C) - широко відомий алгоритм сортування, розроблений англійським інформатики Чарльзом Хоором під час його роботи в МДУ в 1960 році.

Один із самих швидких відомих універсальних алгоритмів сортування масивів: в середньому $O(n \log n)$ обмінів при упорядкуванні n елементів, через наявність деяких недоліків на практиці зазвичай використовується з доробками.

Відмінною особливістю швидкого сортування є операція розбиття масиву на дві частини відповідно опорного елемента. Наприклад, якщо послідовність потрібно впорядкувати по зростанню, то в ліву частину будуть поміщені всі елементи, значення яких менше значення опорного елемента, а в праву елементи, значення яких більше або дорівнюють опорному.

Незалежно від того, який елемент обрано в якості опорного, масив буде відсортовано, найбільш вдалою вважається ситуація, коли по обидва боки від опорного елемента буде приблизно рівна кількість елементів. Якщо довжина якоїсь з частин отриманих в результаті розбиття перевищує один елемент, то для неї потрібно рекурсивно виконати упорядкування, а саме повторно запустити алгоритм на кожному з відрізків.

Таким чином, алгоритм швидкого сортування включає в себе два основних пункта:

- 1) розподіл масиву відносно опорного елемента;
- 2) рекурсивне сортування кожної частини масиву.

Гірший час: $O(n^2)$.

Кращий час: $O(n \log n)$ (звичайний поділ) або $O(n)$ (поділ на 3 частини).

Середній час: $O(n \log n)$.

Витрати пам'яті: $O(n)$ допоміжних, $O(\log n)$ допоміжних.

Реалізація алгоритму в мові програмування C#:

C#:

```
int partition (int[] array, int start, int end)
{
    int marker = start;
    for ( int i = start; i <= end; i++ )
    {
        if ( array[i] <= array[end] )
        {
            int temp = array[marker]; // swap
            array[marker] = array[i];
            array[i] = temp;
            marker += 1;
        }
    }
    return marker - 1;
}

void quicksort (int[] array, int start, int end)
{
    if ( start >= end )
    {
        return;
    }
    int pivot = partition (array, start, end);
    quicksort (array, start, pivot-1);
    quicksort (array, pivot+1, end);
}
```

C# з узагальненими типами, тип T повинен реалізовувати інтерфейс ICOMPARABLE <T>:

IComparable <T> інтерфейс - виявляє узагальнений метод порівняння, який реалізується типом значення або класом для створення метода порівняння з метою упорядкування або сортування екземплярів.

Параметри типу `<T>` - тип об'єктів для порівняння. Це контрваріаційного параметру. Це означає, що ви можете використовувати будь-який із запропонованих типів або будь-який тип, який менш похідний.

Похідний: `System.Boolean`

`System.Byte`

`System.Char`

`System.DateTime`

`System.DateTimeOffset`

`System.Decimal`

`System.Double`

`System.Guid`

`System.Half`

`System.Int16`

`System.Int32`

`System.Int64`

`System.IntPtr`

`System.SByte`

`System.Single`

`System.String`

`System.TimeSpan`

`System.UInt16`

`System.UInt32`

`System.UInt64`

System.UIntPtr

System.ValueTuple

System.ValueTuple<T1>

System.ValueTuple<T1,T2>

System.ValueTuple<T1,T2,T3>

System.ValueTuple<T1,T2,T3,T4>

System.ValueTuple<T1,T2,T3,T4,T5>

System.ValueTuple<T1,T2,T3,T4,T5,T6>

System.ValueTuple<T1,T2,T3,T4,T5,T6,T7>

System.ValueTuple<T1,T2,T3,T4,T5,T6,T7,TRest>

System.Version

System.IO.Log.SequenceNumber

System.Numerics.BigInteger

System.Security.Principal.SecurityIdentifier

System.ServiceModel.Discovery.DiscoveryMessageSequence

System.Text.Rune

int partition<T>(T[] m, int a, int b) where T :IComparable<T>

Алгоритм:

{

int i = a;

for (int j = a; j <= b; j++) // перегибаем с a по b

```

{
    if (m[j].CompareTo( m[b]) <= 0) // якщо елемент m[j] не перевершує m[b],
    {
        T t = m[i];           // міняємо місцями m[j] и m[a], m[a+1], m[a+2] і так далі...
        m[i] = m[j];          // тобто переносимо елементи менше m[b] в початок,
        m[j] = t;             // а потім і сам m[b] «зверху»
        i++;                  // таким чином останній обмін: m[b] и m[i], після чого i++
    }
}

return i - 1;               // в індексі i зберігається <нова позиція елемента m[b]> +
1
}

void quicksort<T>( T[] m, int a, int b) where T : IComparable<T> // a - початок
підмножин, b - кінець

{
    // для першого виклику: a = 0, b = <елементів в масиві> - 1

    if (a >= b) return;

    int c = partition( m, a, b);

    quicksort( m, a, c - 1);

    quicksort( m, c + 1, b);

}

```

Приклади виклику:

- 1) `double[] arr = {9,1.5,34.4,234,1,56.5};`
- 2) `quicksort<double>(arr,0,arr.Length-1);`

С# з використанням лямбда-виразів:

Лямбда-вираз – це будь-який вираз із двох наступних форм:

- Лямбда вираз в якому вираз в якості тексту: `(input-parameters) => expression.`
- Лямбда оператор – має блок операторів в якості тексту:
`(input-parameters) => { <sequence-of-statements> }`

Для оголошення лямбда-виразу використовуйте оператор `=>` для того щоб відокремити список параметрів лямбда-виразів від виконуваного коду. Для того щоб створити лямбда-вираз, необхідно вказати вхідні параметри з лівої сторони лямбда-оператора і блок виразів або операторів з іншої сторони.

Лямбда-вираз з виразом по праву сторону від оператора `=>` називається виразом лямбда. Вирази-лямбди повертають результат виразу і приймають форму вказану раніше.

Лямбда-інструкція нагадує лямбда-вираз, за виключенням того, що інструкції закривають в фігурні дужки. Тіло лямбди оператора може складатися з будь-якої кількості операторів але на практиці використовують не більше двох або трьох. Лямбда-інструкції не можна використовувати для створення дерев виразів.

```
using System;

using System.Collections.Generic;

using System.Linq;

static public class Qsort

{
```

```
public static IEnumerable<T> QuickSort<T>(this IEnumerable<T> list) where T :
    IComparable<T>
```

```
{
    if (!list.Any())
    {
        return Enumerable.Empty<T>();
    }

    var pivot = list.First();

    var smaller = list.Skip(1).Where(item => item.CompareTo(pivot) <=
0).QuickSort();

    var larger = list.Skip(1).Where(item => item.CompareTo(pivot) >
0).QuickSort();

    return smaller.Concat(new[] { pivot }).Concat(larger);
}
```

//(те ж саме, але записане в один рядок, без оголошення змінних)

```
public static IEnumerable<T> shortQuickSort<T>(this IEnumerable<T> list)
where T : IComparable<T>
```

```
{
    return !list.Any() ? Enumerable.Empty<T>() : list.Skip(1).Where(
        item => item.CompareTo(list.First()) <= 0).shortQuickSort().Concat(new[] {
list.First() })
        .Concat(list.Skip(1).Where(item => item.CompareTo(list.First()) >
0).shortQuickSort());
}
```


2.3 Огляд робіт, де розглянуте аналогічне до теми роботи завдання

Під час інформаційного огляду, були розглянуті тренажери студентів ПУЕТ, а саме: Олефіренка В.О., Шульги І. С. Тренажери були розроблені для дистанційного курсу «Алгоритми та структури даних» та «Алгоритми та структури даних».

Тренажер з теми «Сортування включеннями з лінійним та бінарним пошуком» розроблений студентом Олефіренком В.О. Після завантаження програми користувач бачить відображене вікно на якому зображено інформацію про тему, автора, який розробив тренажер, та тематичний теоретичний матеріал. Для того щоб почати роботу, користувач повинен натиснути «Почати тренінг». Далі користувач буде спостерігати подальші питання з сортування одновимірного масиву. Якщо була допущена помилка під час вибору відповіді, то користувачу буде запропонована ще одна спроба. Якщо ж і подальша відповідь буде невдалою, то програма відобразить повідомлення з підказкою.

Тренажер з теми «Рекурсивне породження переставлень», розроблений студентом Шульгою І. С.. Графічний інтерфейс був реалізований за допомогою фреймворків Bootstrapта jQuery. Запрограмований на мові JavaScript, яка взаємодіє з елементами графічного інтерфейсу. Графічний інтерфейс був розроблений за допомогою технології адаптивного дизайну, що коректно відображає елементи інтерфейсу на будь-якому пристрої.

2.4 Позитивні аспекти оглянутих робіт

- Надано більше однієї спроби для відповіді на поставлене питання.
- Різноманітність завдань, що надають змогу навчити студента майже в повному обсязі розібрати тему.
 - Кожен окремий запуск тренажеру надає студентові різні вхідні данні для сортування, що дає змогу бачити, як працює сортування на різних прикладах.
 - Вибір довжини послідовності, для якої виконується сортування.
 - Зручний та зрозумілий інтерфейс.

2.5 Вади розробок з оглянутих робіт

- В відповідях знаходиться велика кількість текстової інформації, що не дає змогу чітко сконцентруватися на виборі правильної.
- Підказки надають багато інформації щодо правильності відповіді.
- Реалізація запуску тренажеру, який виконаний в вигляді веб-сайту. Для студента, який не знає, який з файлів завантажувати – це є проблемою.

2.6 Необхідність та актуальність теми

Необхідність «Швидкого сортування» полягає в покращенні вже існуючих методів сортування за допомогою прямого обміну, таких як : «Сортування бульбашкою» та «Шейкерне сортування» які відомі своєю низькою ефективністю. Принципова відмінність полягає в тому, що в першу чергу відбувається перестановка на найбільшій відстані і після кожного переходу елементи діляться на дві незалежні частини. Таким чином покращення самого неефективного прямого методу сортування дає в результаті один за найбільш ефективних покращених методів.

Хоор розробив цей метод стосовно до машинного перекладу; словник зберігався на магнітній стрічці, і сортування слів оброблюваного тексту дозволяло отримати переклади за один прохід стрічки, без перемотування її назад. Алгоритм був придуманий Хоором під час його перебування в Радянському Союзі, де він навчався в Московському університеті комп'ютерному перекладу та займався розробкою російсько-англійського розмовника.

На даний момент «Швидке сортування» являється одним із найкращих серед існуючих видів сортування через простоту, швидкість та ефективність. Тому на мою думку тема «Швидкого сортування» є актуальною в програмуванні і в інших сферах використання на даний момент.

3 ПРАКТИЧНА ЧАСТИНА

3.1 Алгоритм задачі за темою роботи

Після завантаження програми, яка реалізує тренажер з теми «Сортування Хоора» дисципліни «Алгоритми та структури даних», студенту відкривається вікно з темою та автором. Щоб почати роботу необхідно натиснути кнопку «Старт». Після чого користувачеві будуть відображені запитання та завдання з теми «Сортування Хоора». Після помилки буде надана ще одна спроба на відповідь.

Питання №1. «Ким було розроблено?», та варіанти відповіді:

- Чарльзом Хоором,
- Чарльзом Діккенсом,
- Альбертом Ейнштейном.

Якщо вибраним варіантом відповіді був перший пункт, то тренажер переходить на наступне питання, в інших випадках повідомлення про помилку: «Спробуйте ще».

Питання №2. «В якому році Чарльз Хоор розробив свій метод сортування?» та варіанти відповіді:

- В 1960,
- В 1860,
- В 1970.

Якщо вибраним варіантом відповіді був перший пункт, то тренажер переходить на наступне питання, в інших випадках повідомлення про помилку: «Спробуйте ще».

Питання №3. «Яку альтернативну назву має сортування Хоора?» та варіанти відповіді:

- Bsort,
- Моментальне сортування,
- Qsort.

Якщо вибраним варіантом відповіді був третій пункт, то тренажер переходить на наступне питання, в інших випадках повідомлення про помилку: «Спробуйте ще».

Питання №4. «До яких методів обміну належить сортування Хоора?» та варіанти відповіді:

- До простих,
- До вдосконалених.

Якщо вибраним варіантом відповіді був другий пункт, то тренажер переходить на наступне питання, в інших випадках повідомлення про помилку: «Спробуйте ще».

Питання №5. «Покращеним варіантом якого сортування є сортування Хоора?» та варіанти відповіді:

- Прямого обміну,
- Часткового обміну.

Якщо вибраним варіантом відповіді був другий пункт, то тренажер переходить на наступне питання, в інших випадках повідомлення про помилку: «Спробуйте ще».

Питання №6. «В чому полягає принципова відмінність Qsort від прямого обміну?» та варіанти відповіді:

- Перестановки проводяться на найбільшій відстані.

- Перестановки проводяться на елементах поруч.

Якщо вибраним варіантом відповіді був перший пункт, то тренажер переходить на наступне питання, в інших випадках повідомлення про помилку: «Спробуйте ще».

Питання №7. «Як схематично буде виглядати сортування Хоора?» та варіанти відповіді:

- Перший варіант:

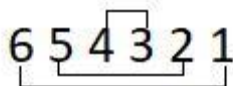


Рисунок 3.1 — Перший варіант

- Другий варіант:

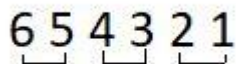


Рисунок 3.2 — Другий варіант

Якщо вибраним варіантом відповіді був перший пункт, то тренажер переходить на наступне питання, в інших випадках повідомлення про помилку: «Спробуйте ще».

Питання №8. «Зі скількох кроків складається загальна ідея алгоритму?» та варіанти відповіді:

- З трьох кроків.
- З двох кроків.

Якщо вибраним варіантом відповіді був перший пункт, то тренажер переходить на наступне питання, в інших випадках повідомлення про помилку: «Спробуйте ще».

Питання №9. «В чому суть першого кроку алгоритму?» та варіанти відповіді:

- Вибрати два елементи, з яких почнеться сортування.
- Вибрати з масиву елемент названий опорним.

Якщо вибраним варіантом відповіді був перший пункт, то тренажер переходить на наступне питання, в інших випадках повідомлення про помилку: «Спробуйте ще».

Питання №10. «В чому суть другого кроку алгоритму?» та варіанти відповіді:

- Порівняти всі інші елементи з опорним.
- Поставити елементи з меншого.

Якщо вибраним варіантом відповіді був другий пункт, то тренажер переходить на наступне питання, в інших випадках повідомлення про помилку: «Спробуйте ще».

Питання №11. «Під час другого кроку на скільки неперервних відрізків розбивають масив?» та варіанти відповіді:

- На три відрізка.
- На два відрізка.

Якщо вибраним варіантом відповіді був другий пункт, то тренажер переходить на наступне питання, в інших випадках повідомлення про помилку: «Спробуйте ще».

Питання №12. «Які неперервні відрізки створюються під час другого кроку?» :

- «Менші опорного», «рівні», «більші».
- «Менші опорного», «більші опорного».

Якщо вибраним варіантом відповіді був перший пункт, то тренажер переходить на наступне питання, в інших випадках повідомлення про помилку: «Спробуйте ще».

Питання №13. «Що потрібно зробити третім кроком?» :

- Для відрізка «більших» і «менших» значень зробити рекурсивну послідовність операцій.
- Для відрізка «менших» значень зробити послідовну операцію.

Якщо вибраним варіантом відповіді був перший пункт, то тренажер переходить на наступне питання, в інших випадках повідомлення про помилку: «Спробуйте ще».

Питання №14. «На скільки частин ділять масив для зручності на практиці?» :

- На чотири частини.
- На дві частини.

Якщо вибраним варіантом відповіді був другий пункт, то тренажер переходить на наступне питання, в інших випадках повідомлення про помилку: «Спробуйте ще».

3.2 Розробка блок-схеми алгоритму, який підлягає програмуванню

Блок-схема- поширений тип графічних моделей, описуючих алгоритми або процеси в яких окремі кроки зображують у вигляді блоків різних форм, поєднаних між собою лініями вказуючими напрямком послідовності.

Обмежувач – символ відображає вхід із зовнішнього середовища і вихід в зовнішнє середовище, тобто начало або кінець схеми програми, зовнішнє використання і джерело або пункт призначення даних.

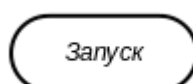


Рисунок 3.3 — Приклад символу обмежувача в блок-схемі.

На практиці має сенс наступний запис обмежувача в блок-схемі: початок\кінець, перезапуск(перезапуск даної блок-схеми), помилка (завершення алгоритму з помилкою). Нижче на рисунку 3.4 зображено приклад обмежувача в блок-схемі.

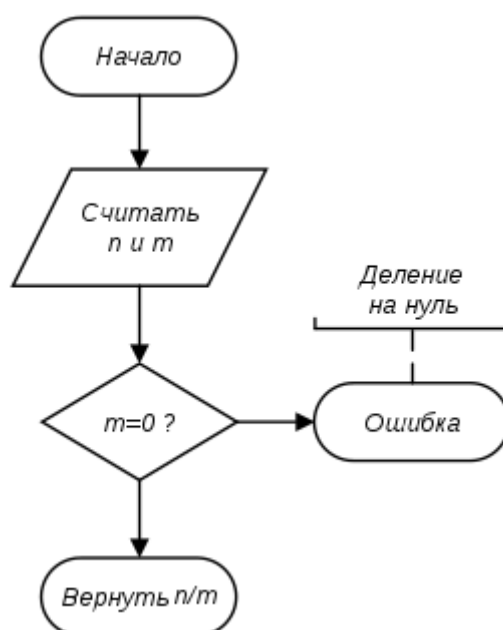


Рисунок 3.4 — Приклад символу обмежувача в блок-схемі.

Коментар – символ використовують для додання описових коментарів або пояснювальних записів в цілях пояснення або приміток. Пунктирні лінії в символі коментаря зв’язані з відповідним символом або можуть обводити групу символів. Текст коментаря повинен бути поміщений біля обмежуючої фігури, як зображено на рисунку 3.5.

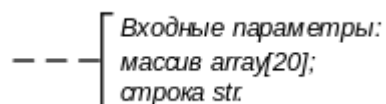


Рисунок 3.5 — Приклад символу коментаря.

Питання (умова або вирішення) – символ відображає рішення або функцію перемикаючого типу, маючи один вхід і декілька альтернативних виходів, один і тільки один з яких може бути активований після рішення умови, вказаних всередині цього символу. Приклад символу вказано на рисунку 3.6.

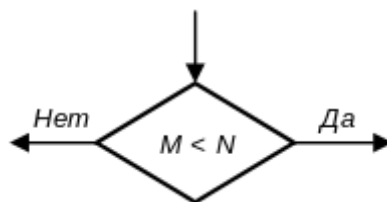


Рисунок 3.6 — Приклад символу питання.

Дія – символ відображає функцію обробки даних будь-якого виду (виконання певної операції або групи операцій, приводить до зміни значень, форми або розміщення інформації або вибору по якому з декількох напрямів потоку слід рухатися). Приклад зображено на рисунку 3.7.



Рисунок 3.7 — Приклад символу дії.

Дані (ввід-вивід) – символ відображає дані, косій не визначений. Перетворення даних в форму, пригідну для обробки або відображення результатів обробки. Даний символ не визначає носія даних (для вказання типу носія даних

використовуються специфічні символи). Приклад символу даних зображено на рисунку 3.8.

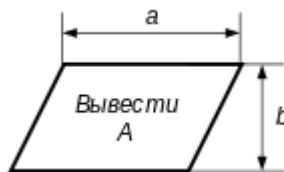


Рисунок 3.8 — Приклад символу даних.

Зумовлений процес (функція) — символ відображає зумовлений процес, складений з однієї або декількох операцій чи кроків програми, котрі зумовлені в іншому місці (програмі, модулі). Наприклад в програмуванні – виклик процедури або функції. Приклад зображено на рисунку 3.9.

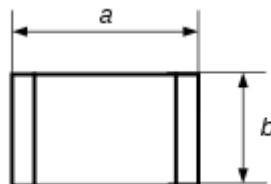


Рисунок 3.9 — Приклад символу функції.

На рис. 3.10-3.11 показано блок-схеми алгоритму навчального тренажера з сортування Хоора. На рис. 3.10 — показана блок-схема алгоритму всього навчального тренажера. На рис. 3.11 зображено блок-схему питання з однією правильною відповіддю.

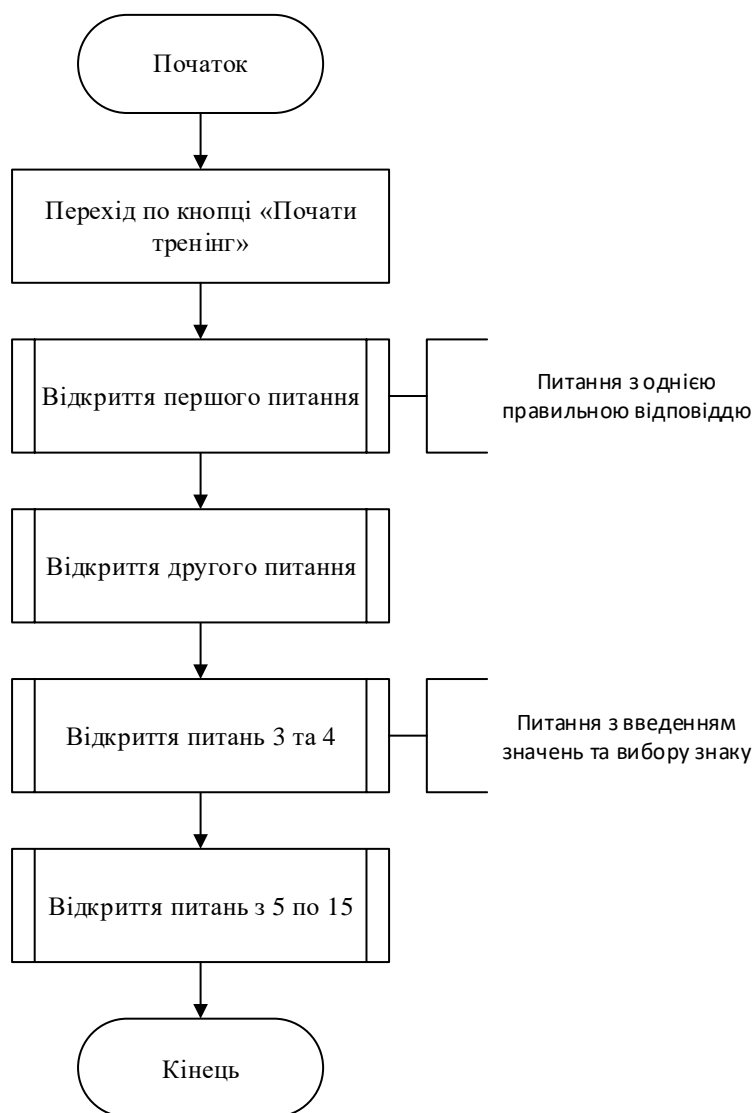


Рисунок 3.10 — Блок-схема всього алгоритму навчального тренажера

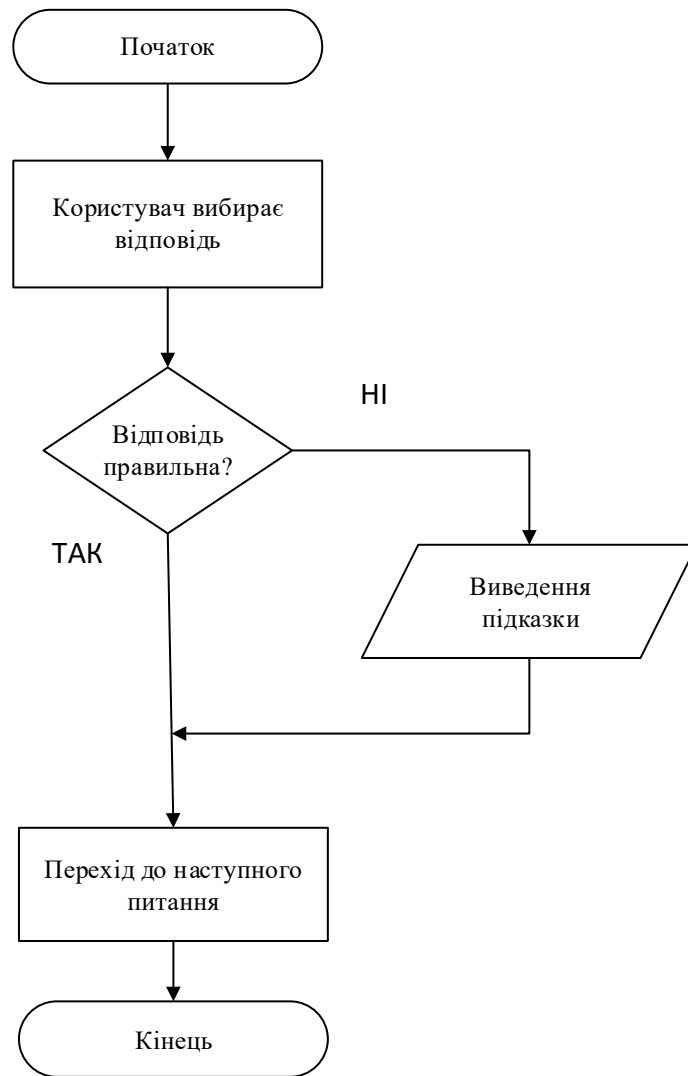


Рисунок 3.11 — Блок-схема питання з однією відповіддю

3.3 Обґрунтування вибору програмних засобів для реалізації завдання роботи

Для реалізації програми обрав мову програмування С#, відповідно, платформу .NET, інтегроване середовище розробки Microsoft Visual Studio 2019.

Обрав платформу .NET через:

- .NET ґрунтується на єдиній об'єктно-орієнтованій моделі;
- програма розроблена з допомогою .NET, не залежить від інших програм та від ОС;
- програму можна встановити звичайним копіюванням файлів;
- використання безпечних типів даних, що підвищує надійність програми та сумісність;
- програма взаємодіє з єдиною моделлю обробки помилок;
- зручно повторно використовувати код;

Також є деякі недоліки, які не впливають на досягнення поставлених вимог та остаточного результату:

- наявність необхідної бібліотеки Framework.

4 ПРАКТИЧНА ЧАСТИНА

4.1 Опис процесу програмної реалізації

Спочатку створив проект на мові C#, додаток Windows FormsApplication додаючи проекту форми Windows Form, як зображено на рисунку 4.1 та 4.2.

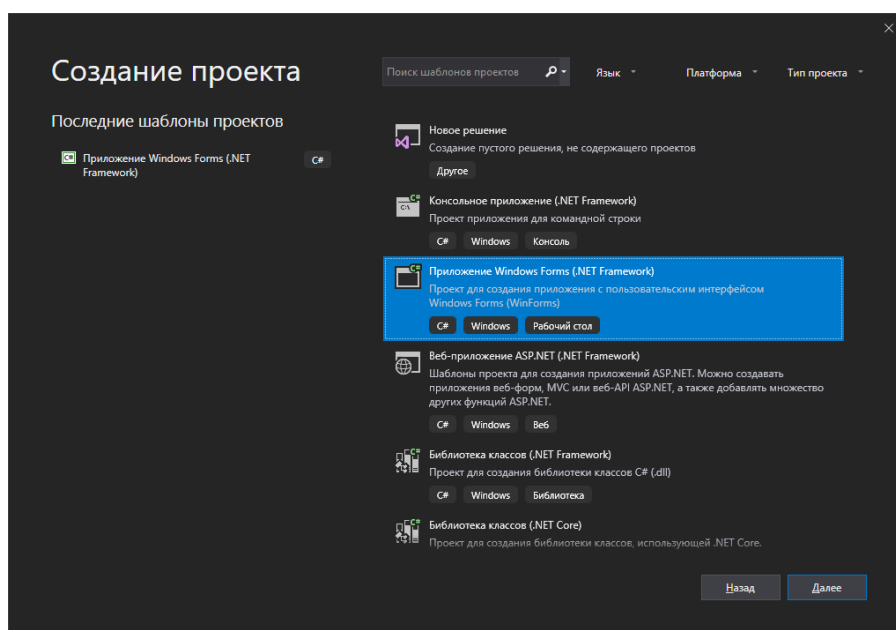


Рисунок 4.1 - Створення проекту

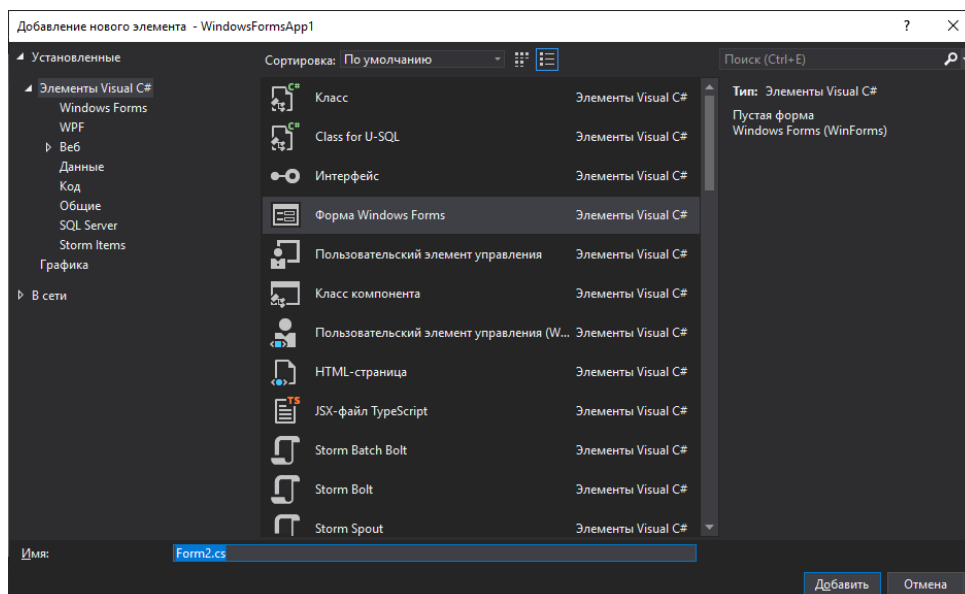


Рисунок 4.2 – Створення нової форми

Після того, як створив першу форму, я назвав її «Тренажер», а далі назва кожної форми відповідала номеру запитання, яке було в ній. Під час створення тренажеру були використанні такі функції, як: Button, Label, RadioButton, та PictureBox, TextBox. Кожен із елементів був використаний по потребі в кожній із форм.

В початковому тренажері є вибір відповіді, тож їх правильність реалізована через оператор *if*. На рисунку 4.3 показана перевірка вибору відповіді.

```
if (radioButton1.Checked)
{
    fr3.Show();
    Hide();
}
else
{
    MessageBox.Show("Спробуйте ще!", "Повідомлення!",
        MessageBoxButtons.OK,
        MessageBoxIcon.Information,
        MessageBoxDefaultButton.Button1);
}
```

Рисунок 4.3 – Перевірка вибору відповіді з одним варіантом

Для зручності під час проходження тренінгу було реалізовано вікно-повідомлення (рисунок 4.4), що повідомляє про неправильність відповіді, якщо відповідь була такою.

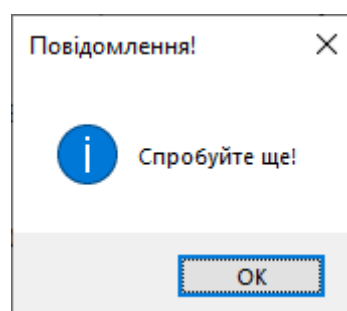


Рисунок 4.4 – Вікно, що з'являється у випадку невірної відповіді

4.2 Опис програми

Після запуску програми екрані відображено вікно, де вказана назва роботи, «Тренажер множин для дистанційного курсу «Алгоритми та структури даних»», кнопку «Старт». Після натиску на кнопку «Старт» починається тестування. На рисунку 4.6 зображена перша форма.

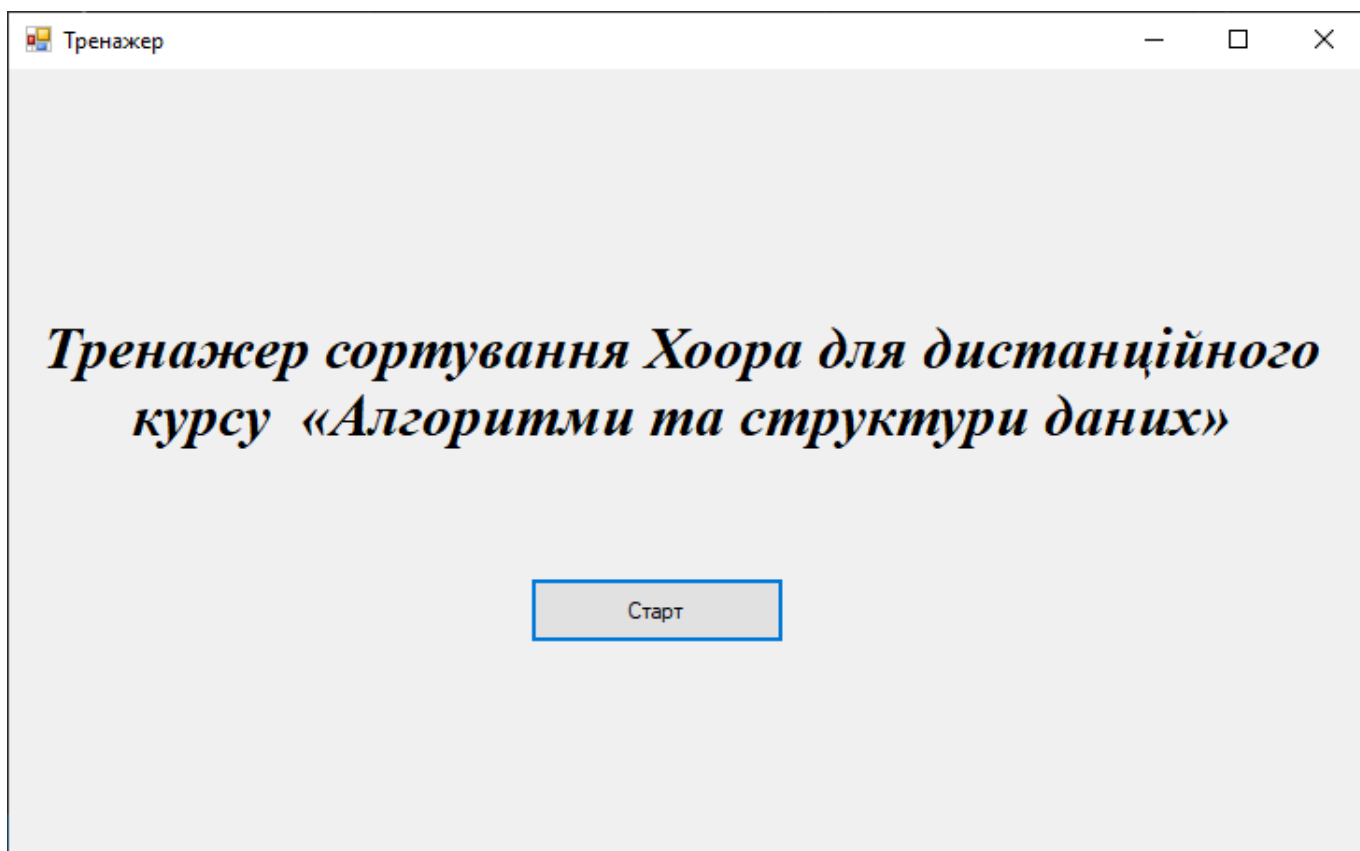


Рисунок 4.6 – Стартовий екран

Після переходу на наступний екран з вибором правильної відповіді. Після вибору свого варіанту відповіді та натиску на кнопку «Ок», користувач або перейде, до іншого запитання, або якщо його відповідь була не вірна, то на екрані з'явиться повідомлення, про помилку .

З прикладом відображення запитань можна ознайомитися, якщо переглянути

рис. 4.7

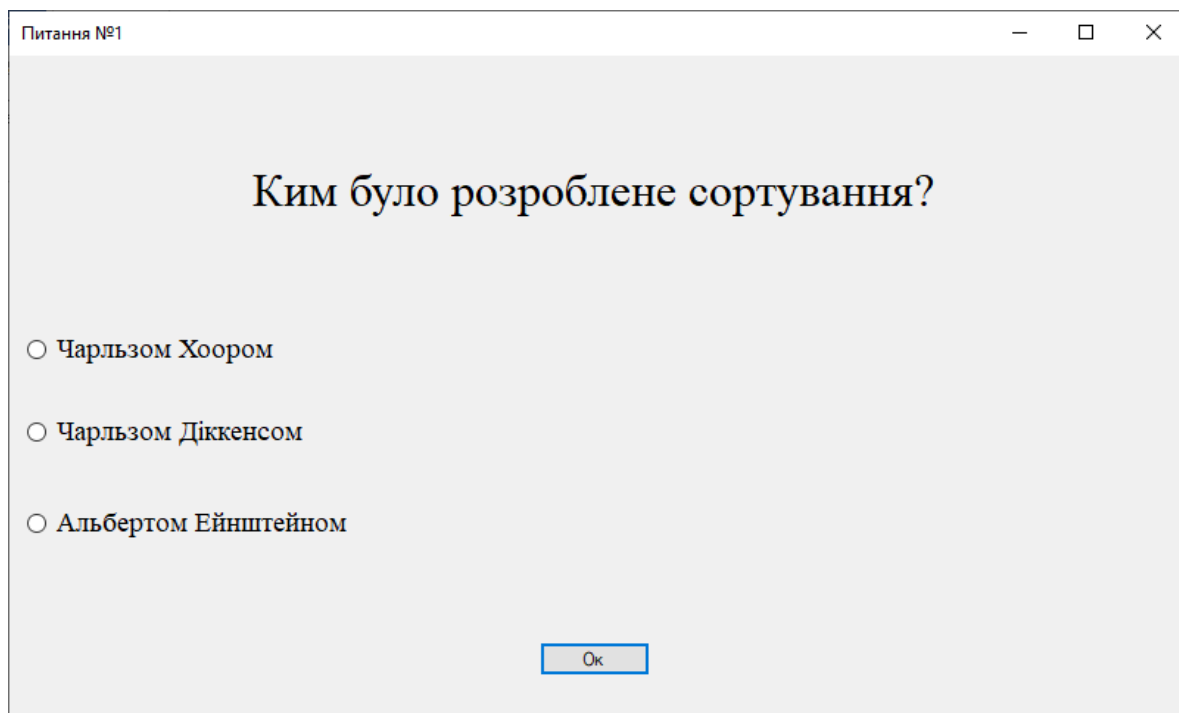


Рисунок 4.7 – Екран з вибором відповіді

4.3 Перевірка валідності

Перевіримо працездатність роботи програми, а саме:

- працездатність в кнопок,
- відображення повідомлення про помилку,
- перехід на наступне питання після правильної відповіді.

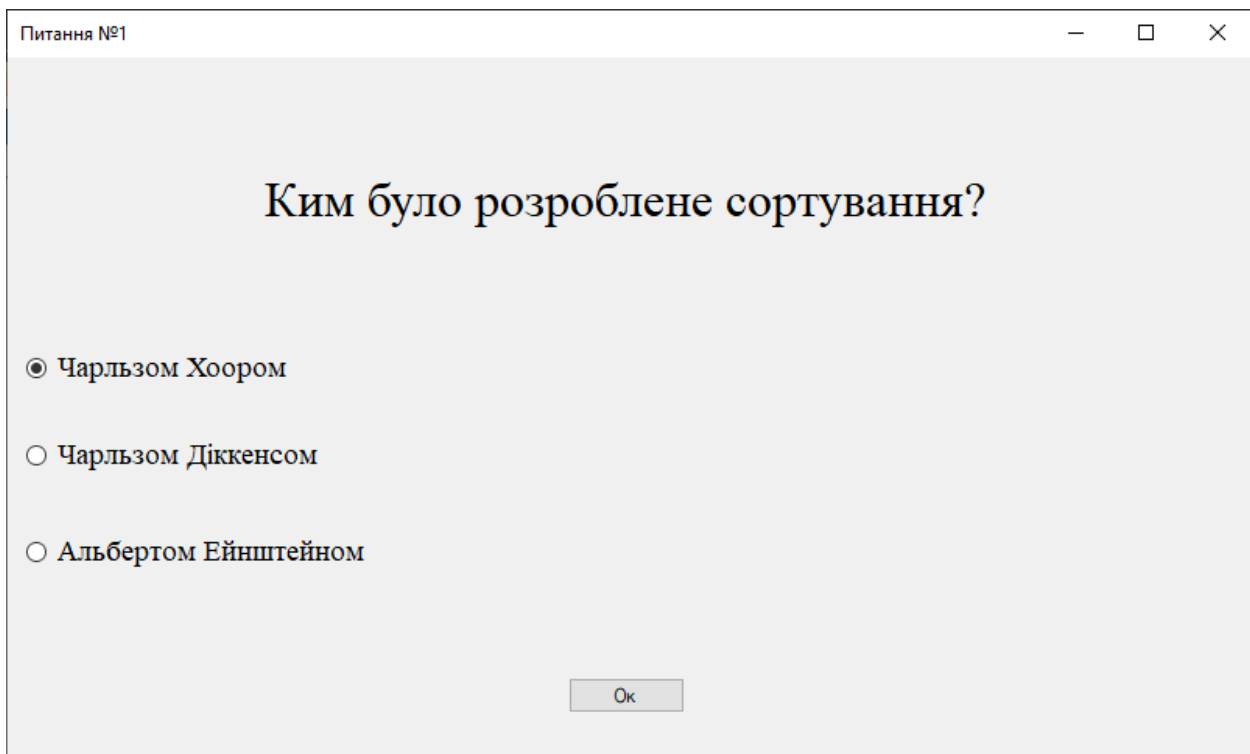


Рисунок 4.8 — Працездатність першого питання.

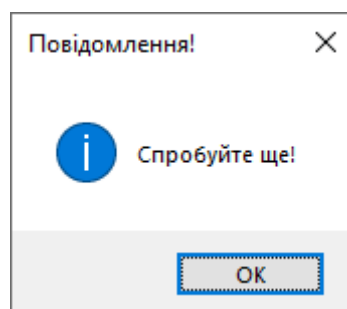


Рисунок 4.9 — Відображення про помилку на першому питанні.

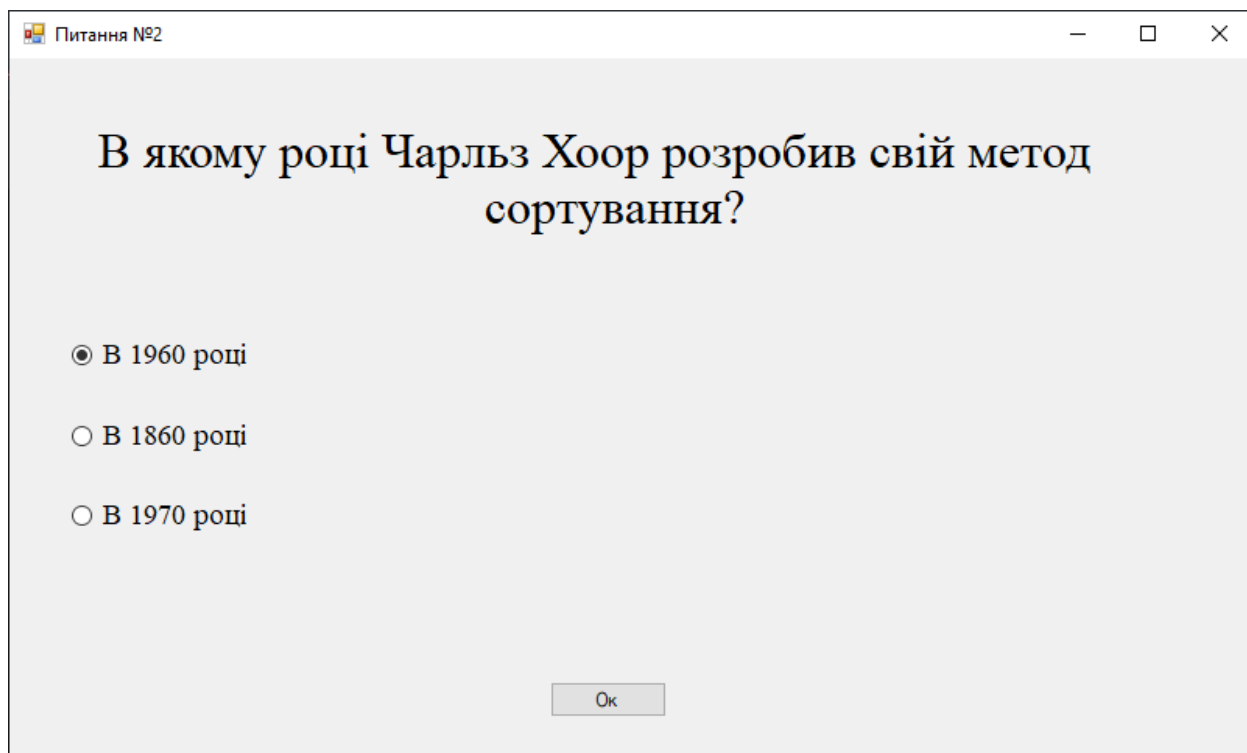


Рисунок 4.10 — Працездатність другого питання.

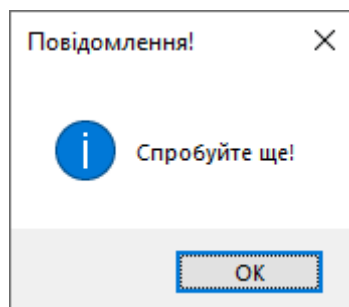


Рисунок 4.11 — Відображення про помилку на другому питанні.

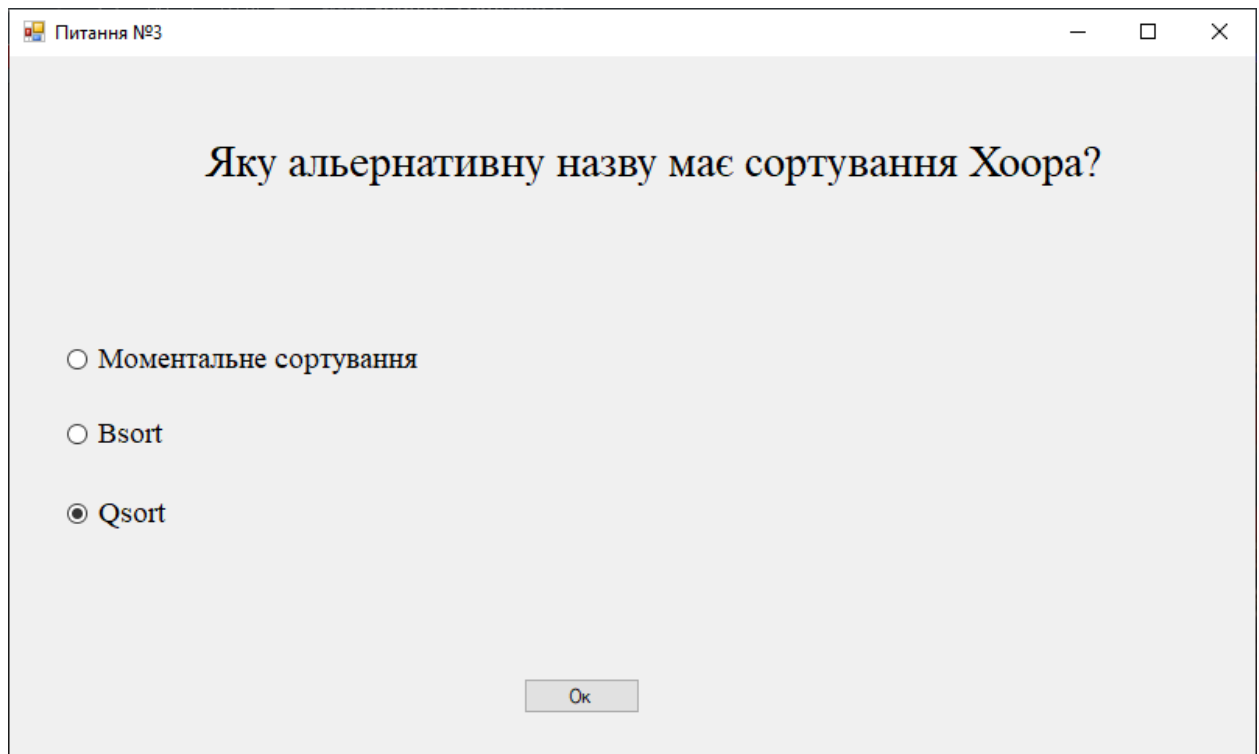


Рисунок 4.12 — Працездатність третього питання.

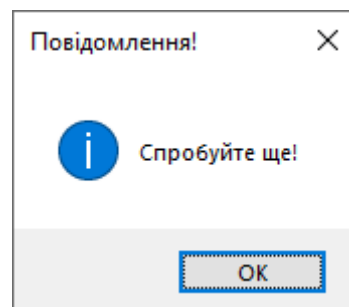


Рисунок 4.13 — Відображення про помилку на третьому питанні.

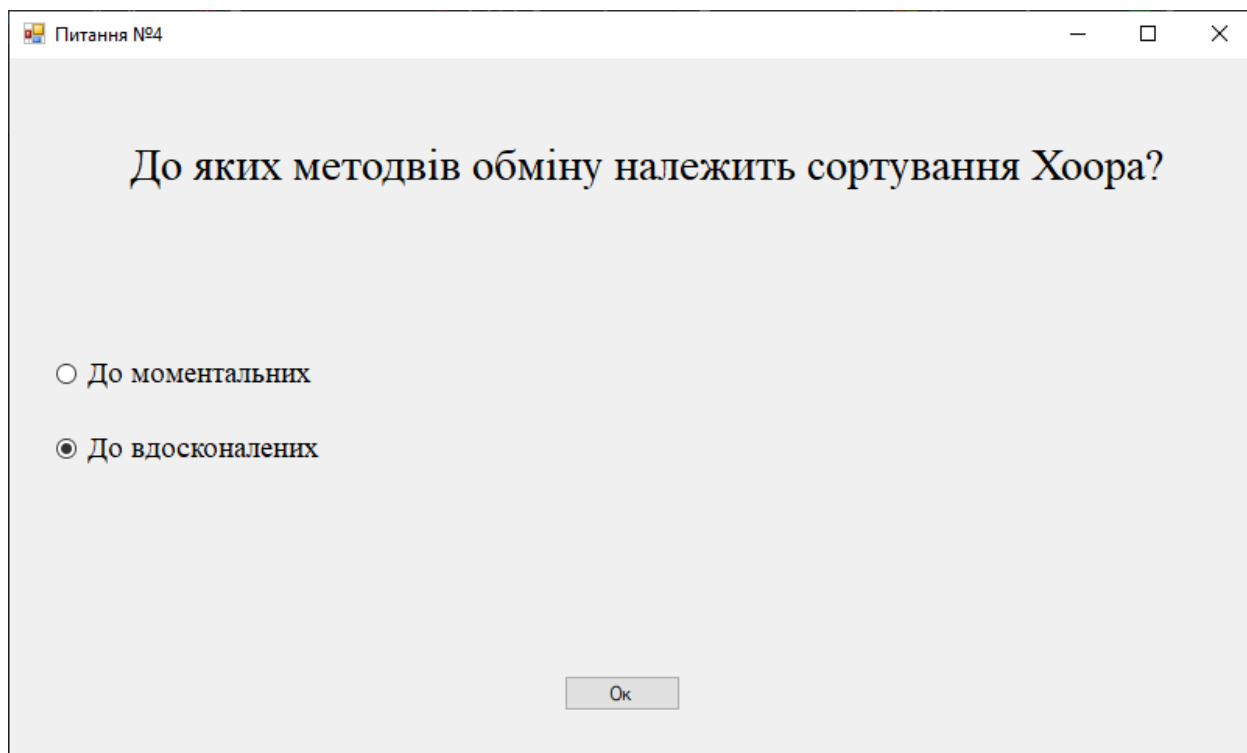


Рисунок 4.14 — Працездатність четвертого питання.

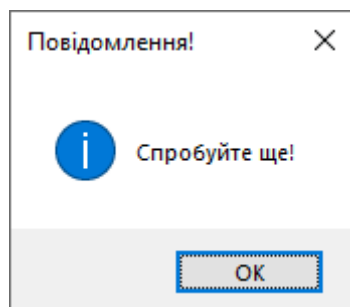


Рисунок 4.15 — Відображення про помилку на четвертому питанні.

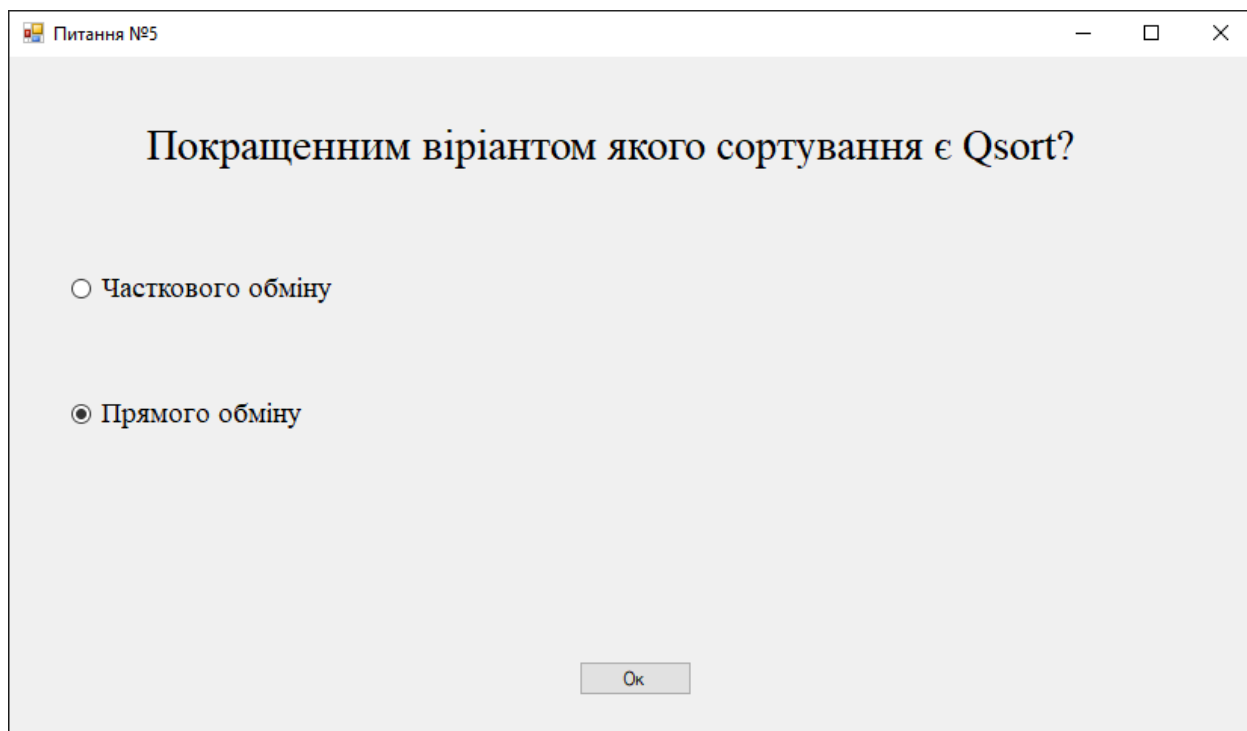


Рисунок 4.16 — Працездатність п'ятого питання.

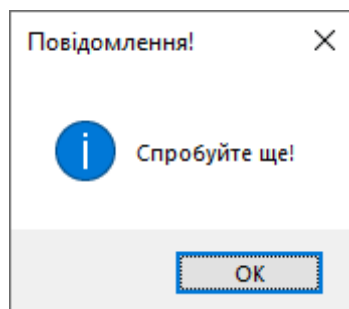


Рисунок 4.17 — Відображення про помилку на п'ятому питанні.

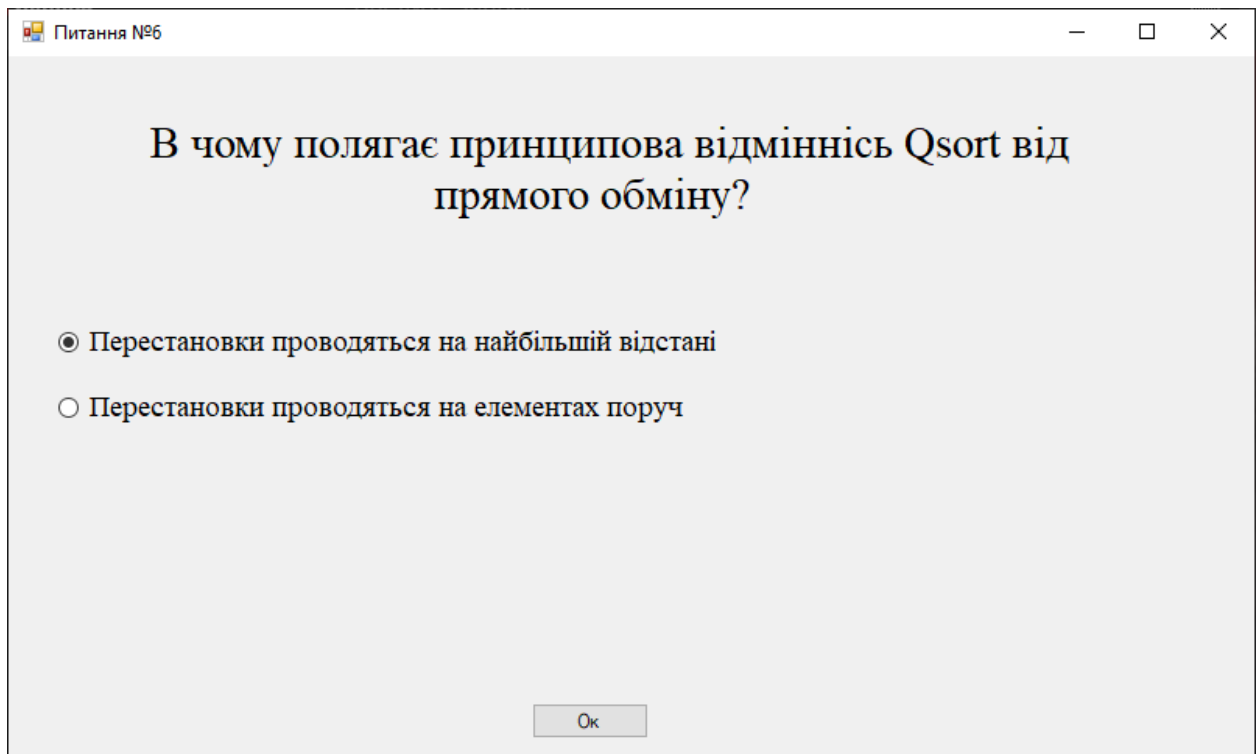


Рисунок 4.18 — Працездатність шостого питання.

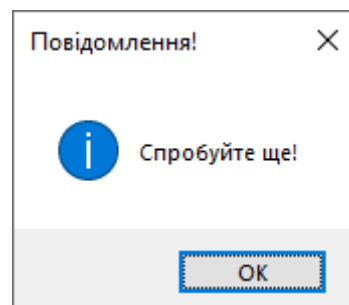


Рисунок 4.19 — Відображення про помилку на шостому питанні.

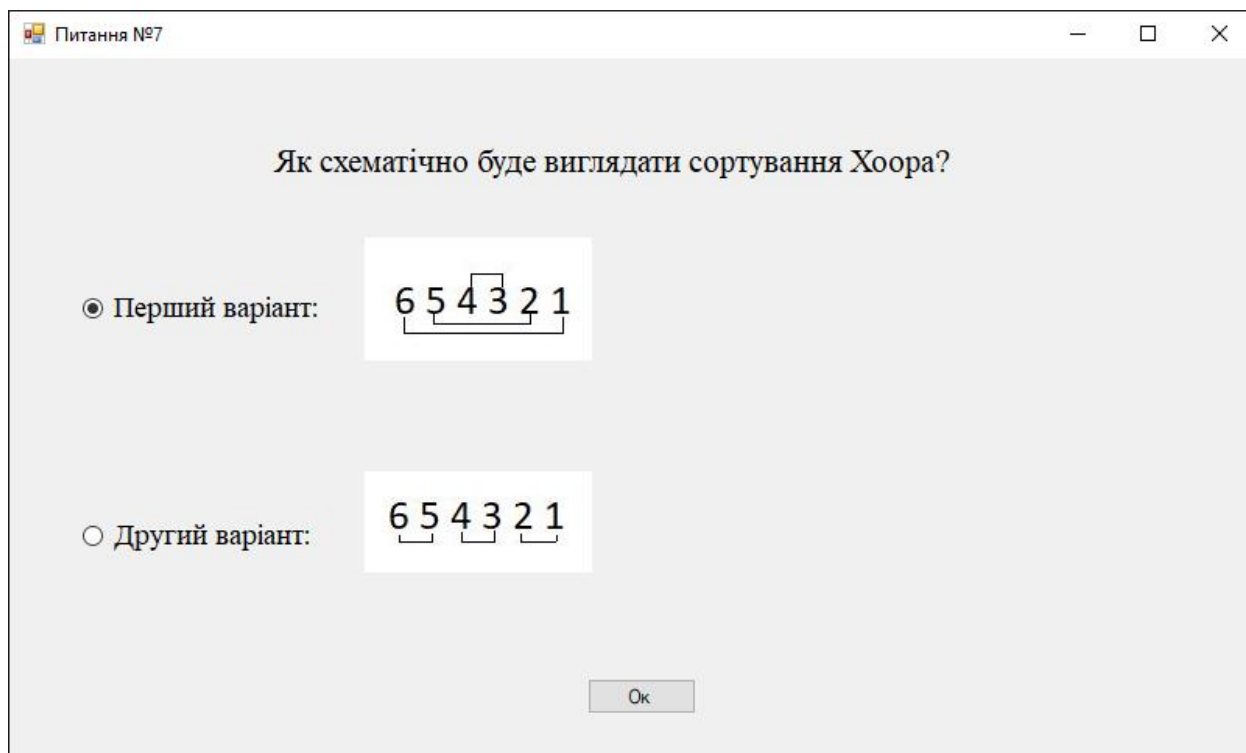


Рисунок 4.20 — Працездатність сьомого питання.

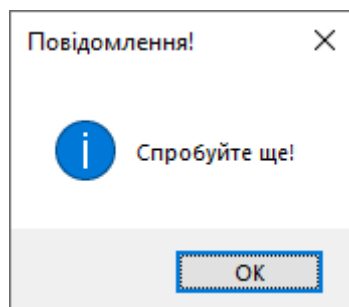


Рисунок 4.21 — Відображення про помилку на сьомому питанні.

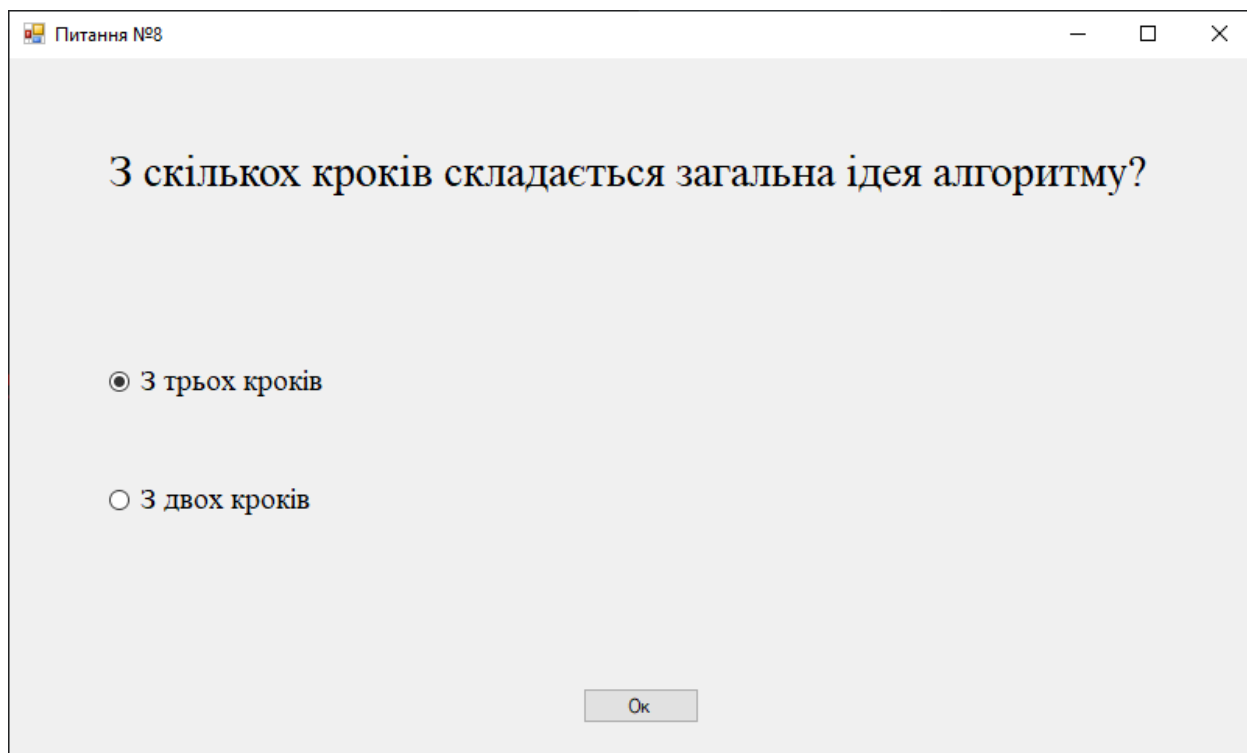


Рисунок 4.22 — Працездатність восьмого питання.

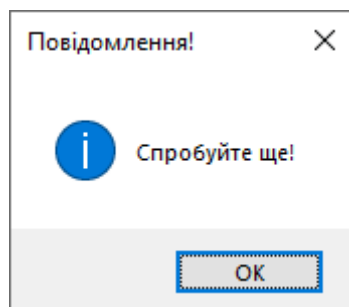


Рисунок 4.23 — Відображення про помилку на восьмому питанні.

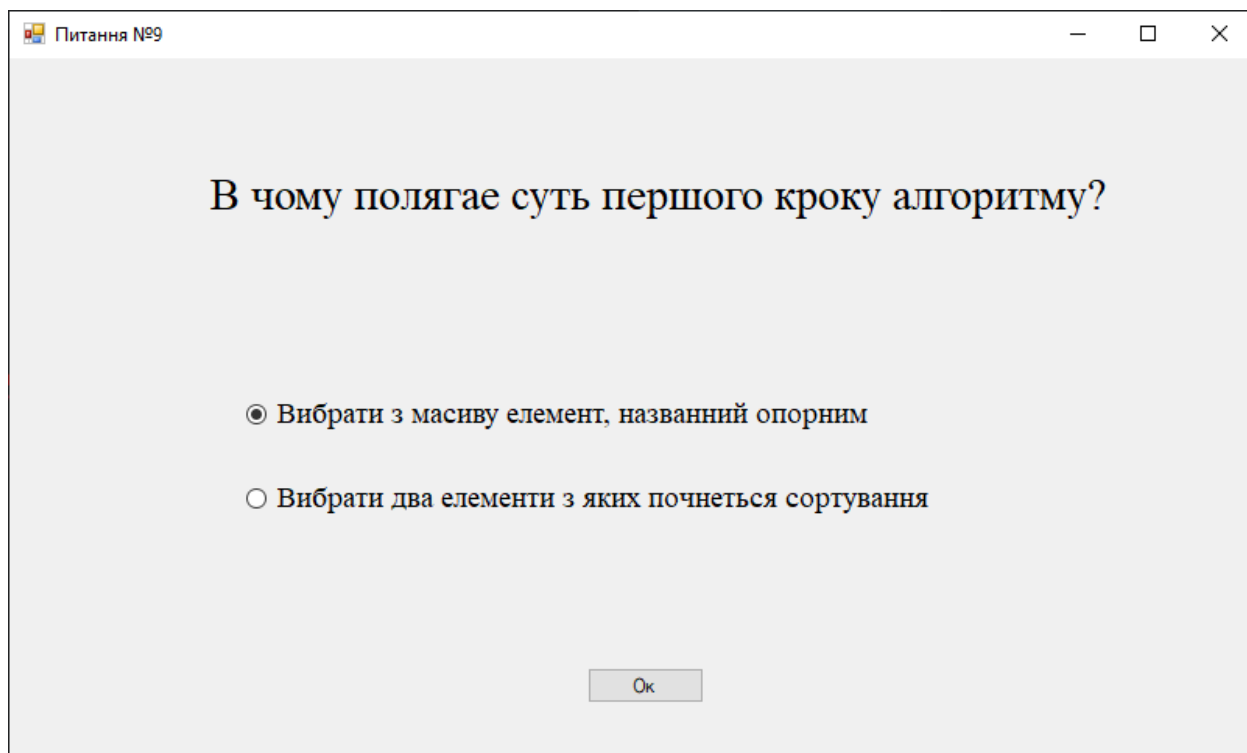


Рисунок 4.24 — Працездатність дев'ятого питання.

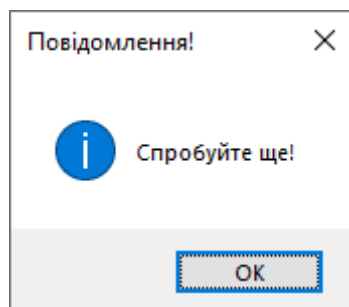


Рисунок 4.25 — Відображення про помилку на дев'ятому питанні.

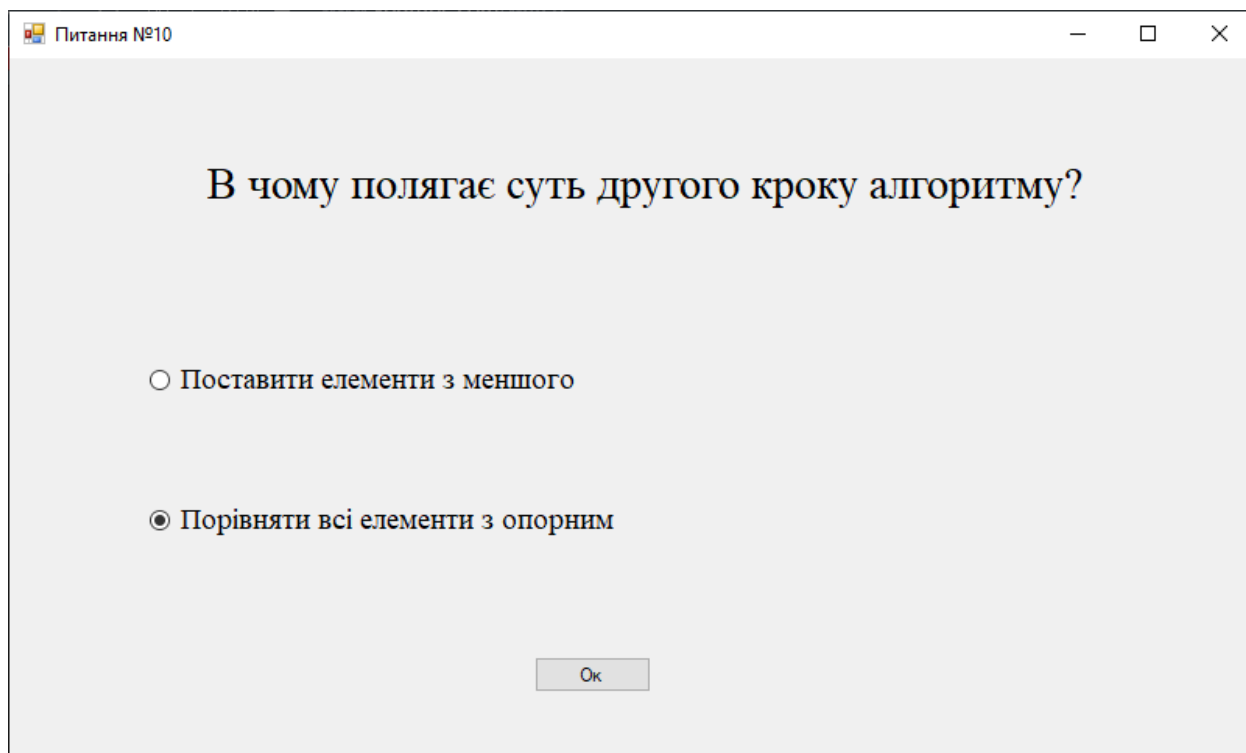


Рисунок 4.26 — Працездатність десятого питання.

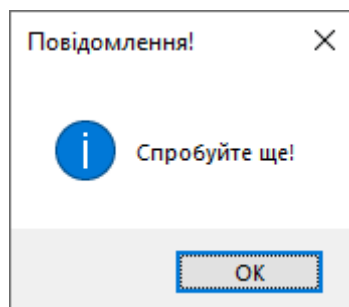


Рисунок 4.27 — Відображення про помилку на десятому питанні.

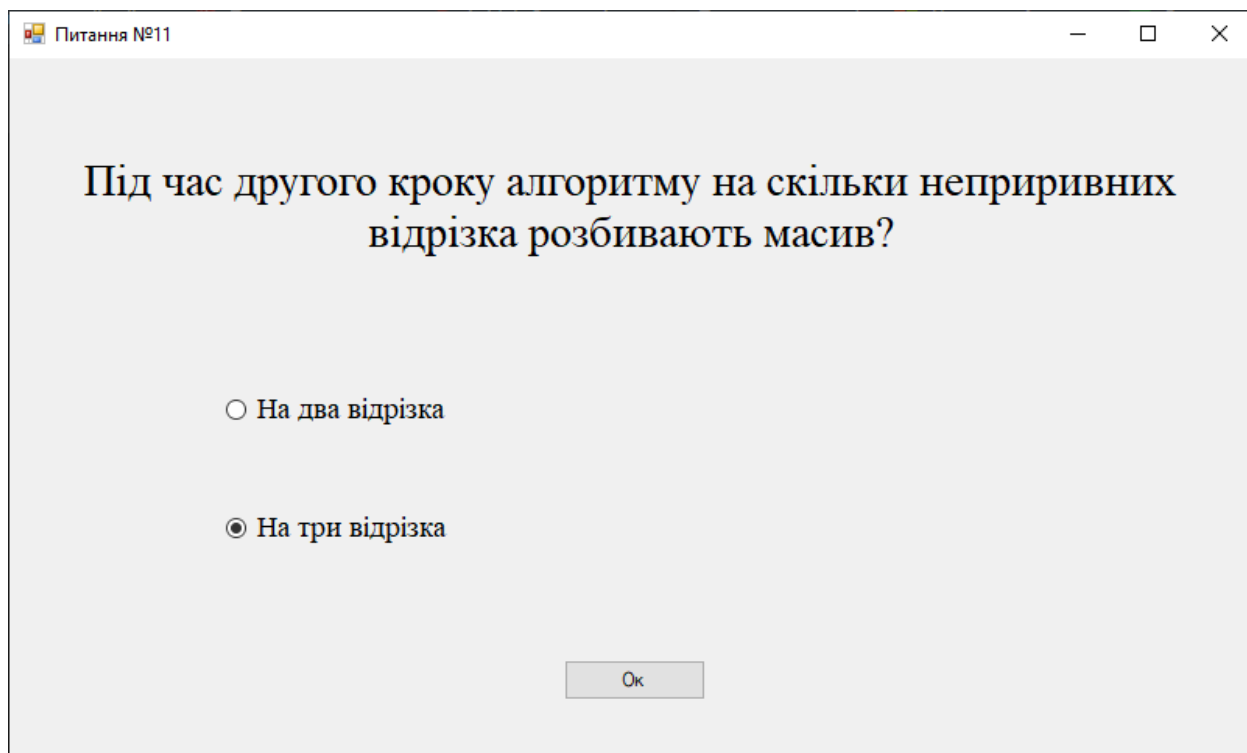


Рисунок 4.28 — Працездатність одинадцятого питання.

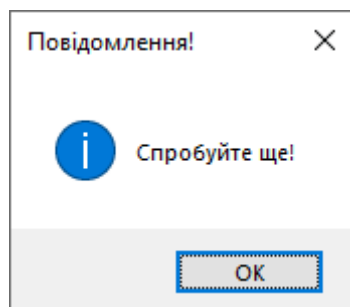


Рисунок 4.29 — Відображення про помилку на одинадцятому питанні.

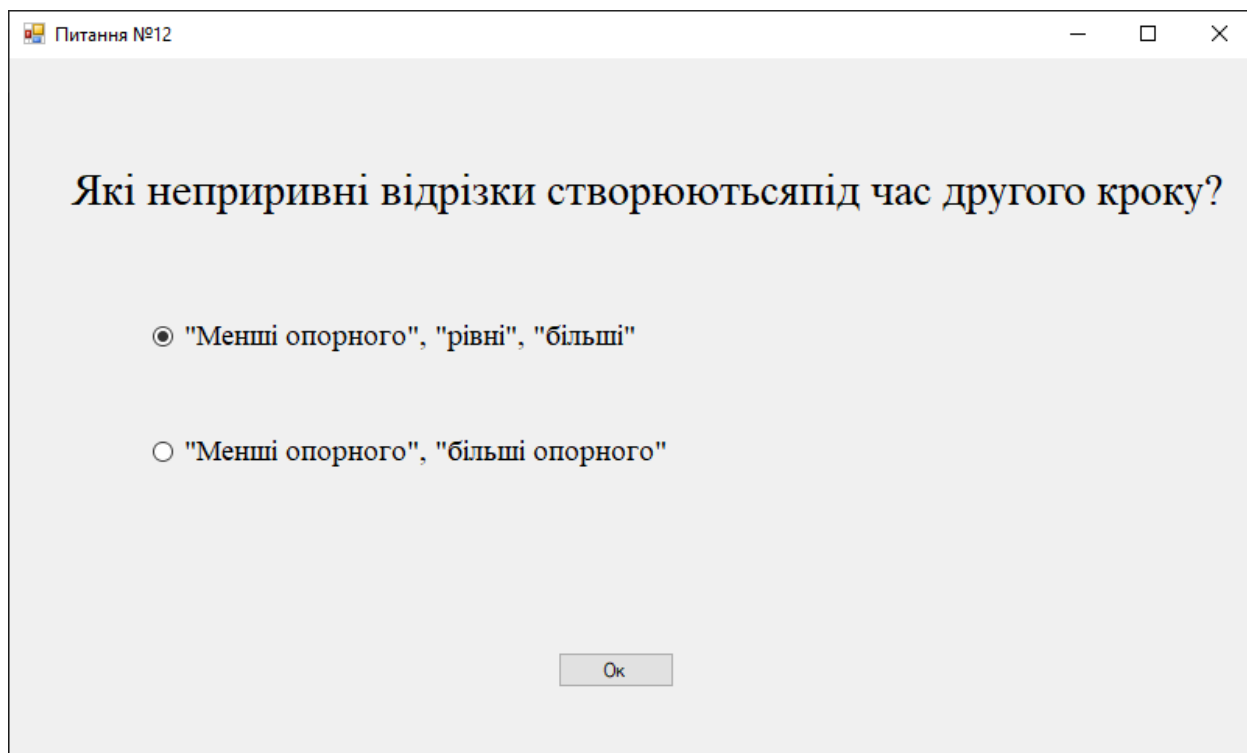


Рисунок 4.30 — Працездатність дванадцятого питання.

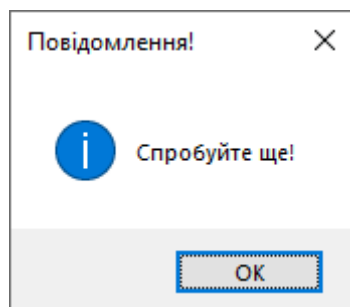


Рисунок 4.31 — Відображення про помилку на дванадцятому питанні.

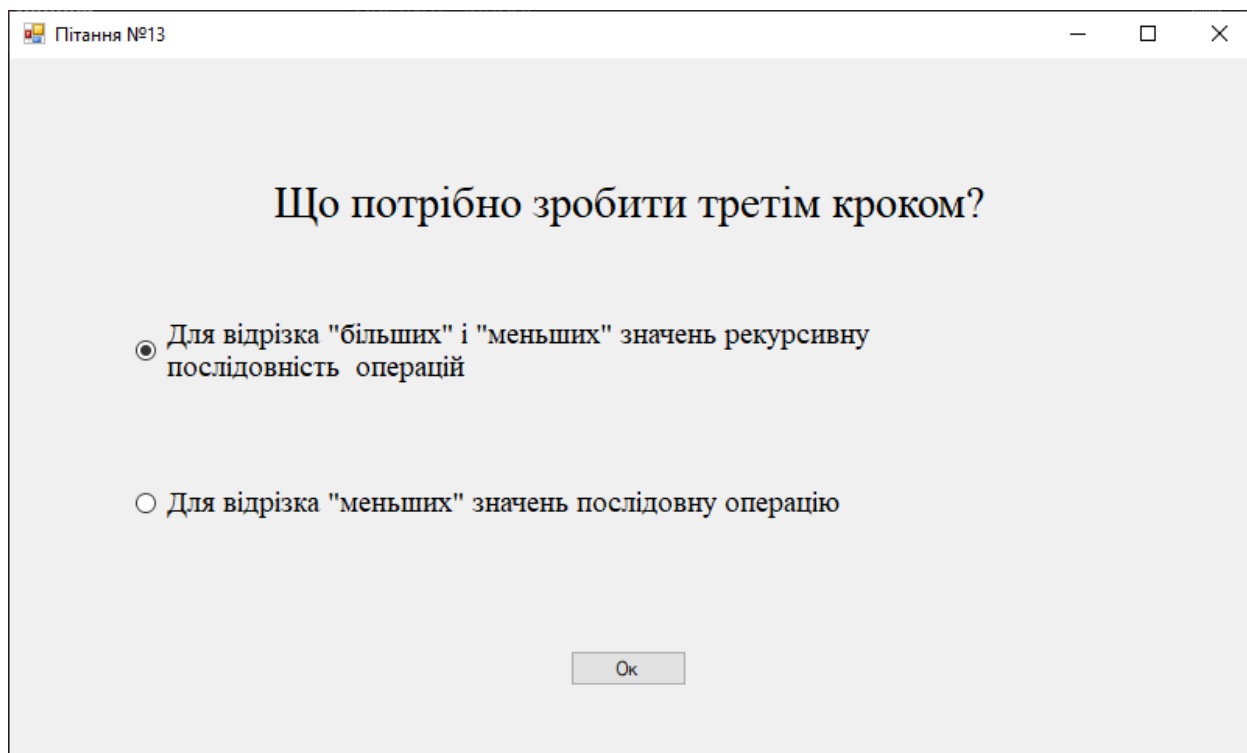


Рисунок 4.32 — Працездатність тринадцятого питання.

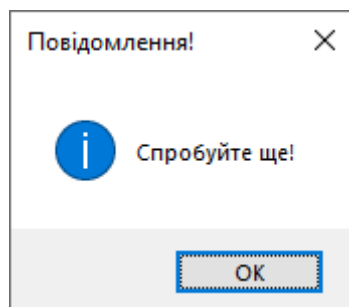


Рисунок 4.33 — Відображення про помилку на тринадцятому питанні.

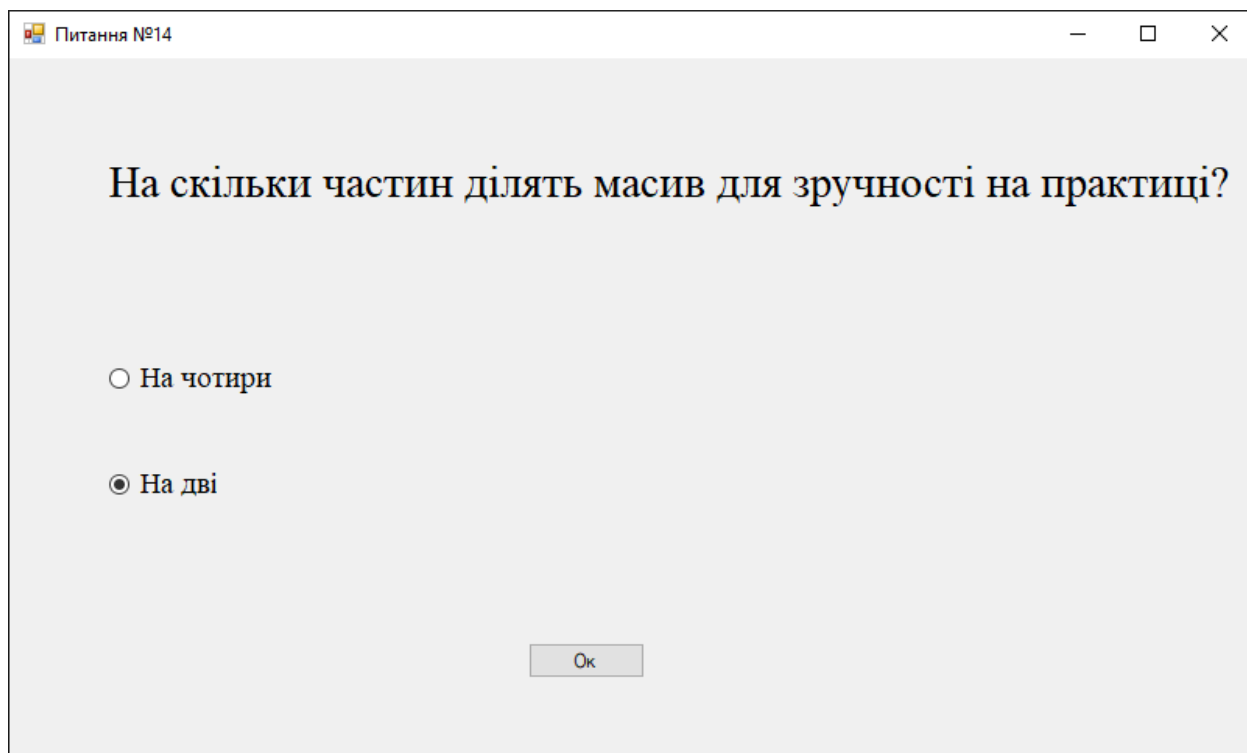


Рисунок 4.34 — Працездатність чотирнадцяте питання.

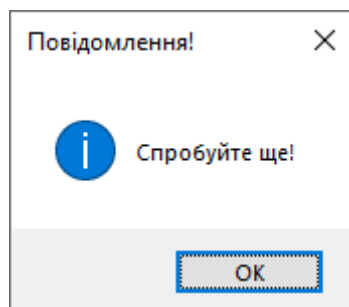


Рисунок 4.35 — Відображення про помилку на чотирнадцятому питанні.

ВИСНОВКИ

Виконуючи магістерську роботу, що ґрунтується на лекціях курсу «Алгоритми та структури даних», розроблено алгоритм та створено навчальний тренажер для сортування Хоора.

Навчальний тренажер не впроваджено в дистанційне навчання ПУЕТ в дистанційний курс «Алгоритми та структури даних» студентів спеціальності «Комп'ютерні науки».

Ознайомився з вимогами для розробки тренажера з теми «Сортування Хоора» для дистанційного курсу «Алгоритми та структури даних»:

1. Обрати мову програмування для створення тренажера. Мова повинна бути сумісна з платформою Moodle.
2. Створити алгоритм тренажера з теми «Сортування Хоора».
3. Реалізувати тренажер за допомогою обраної мови програмування .
4. Перевірити програму на наявність помилок.

Основні вимоги до навчального тренажера:

1. Умова задачі повинна бути доступною для студента на кожному кроці.
2. Після вибору відповіді потрібно реалізувати перевірку даних. Якщо було допущено помилку, то треба повідомити користувача про неї.

Дотримання розглянутих вимог забезпечило коректне функціонування навчального тренажера.

Для реалізації програми було використано мову програмування C#, відповідно, платформу .NET, інтегроване середовище розробки Microsoft Visual Studio 2019.

Обрана платформа .NET через:

- .NET ґрунтується на єдиній об'єктно-орієнтованій моделі;
- програма розроблена з допомогою .NET, не залежить від інших програм та від ОС;
- програму можна встановити звичайним копіюванням файлів;

- використання безпечних типів даних, що підвищує надійність програми та сумісність;
- програма взаємодіє з єдиною моделлю обробки помилок;
- зручно повторно використовувати код;

Також є деякі недоліки, які не впливають на досягнення поставлених вимог та остаточного результату:

- наявність необхідної бібліотеки `FrameWork`.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Концепція розвитку дистанційної освіти в Україні (затверджено Постановою МОН України В. Г. Кременем 20 грудня 2000 р.).
2. Кудрявцева С. П. Міжнародна інформація: навчальний посібник / С. П. Кудрявцева, В. В. Колос. — К.: Видавничий дім «Слово», 2005. — 400 с.
3. Голубенко В. О. Програмна реалізація елементів тренажеру з теми «Сортування бульбашками» дисципліни «Аналіз алгоритмів» / В. О. Голубенко // Комп'ютерні науки і прикладна математика (КНіПМ-2018): матеріали науково-практичного семінару. Випуск 1 / за ред. Ємця О. О. — Полтава: Кафедра ММСІ ПУЕТ, 2018. — С. 11-16 — Режим доступу: <http://dspace.puet.edu.ua/handle/123456789/6563>
4. Русін В. С. Програмна реалізація елементів тренажеру з теми «Аналіз алгоритму сортування вставками» дисципліни «Аналіз алгоритмів» / В. С. Русін // Інформатика та системні науки (ІСН-2017): матеріали VIII Всеукраїнської науково-практичної конференції за міжнародною участю (м. Полтава, 16-18 березня 2017 р.) / за ред. Ємця О. О. — Полтава: ПУЕТ, 2017. — С. 236-237 — Режим доступу: <http://dspace.puet.edu.ua/handle/123456789/5798>
5. Ємець О. О. Методичні рекомендації до виконання кваліфікаційної бакалаврської роботи для студентів 4 курсу напрямку підготовки «Інформатика» / О. О. Ємець, Ол-ра О. Ємець — Полтава: РВВ ПУЕТ, 2017. — 73 с.
6. Ємець Є. М. Візуалізатори алгоритмів сортувань / Є.М. Ємець, Ол-ра О. Ємець, Г.В. Карнаухова // Економіка сьогодні: проблеми моделювання та управління: матеріали IV Всеукр. наук.-практ. Інтернет-конф. (м. Полтава, 15-17 грудня 2014 р.). — Полтава: ПУЕТ, 2015. — С. 61-68.
7. Хрупа О. І. Розробка програмного забезпечення з теми «Турнірне сортування» дистанційного навчального курсу «Алгоритми та структури даних» / О. І. Хрупа, О. О. Ємець // Комп'ютерні науки і прикладна математика (КНіПМ-2019): матеріали наук.-практ. семінару. Випуск 3. / За ред. Ємця О. О. — Полтава: Кафедра

ММСІ ПУЕТ, 2019. – С. 43-45. – Режим доступу: <http://dspace.puet.edu.ua/handle/123456789/7039>.

8. Козодуб В. С. Тренажер «Сортування фон Неймана» / В. С. Козодуб, О. О. Ємець // Комп'ютерні науки і прикладна математика (КНіПМ-2020): матеріали наук.-практ. семінару. Випуск 5. / За ред. Ємця О. О. – Полтава: Кафедра ММСІ ПУЕТ, 2020. – С. 8-11. – Режим доступу: <http://dspace.puet.edu.ua/handle/123456789/8267>.

9. Ємець О.О. Дистанційний курс Полтавського університету економіки і торгівлі «Алгоритми і структури даних» для студентів спеціальності «Комп'ютерні науки» / О.О. Ємець. – [Електронний ресурс].